



T.C

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ MÜHENDİSLİK
VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

GEOMETRİK PROBLEMLER

Hazırlayanlar:

Hilal AYDIN 220501030

Zehra KANDAZ 220501026

DERS SORUMLUSU:

PROF. DR. TARIK DURU

02.01.2024

İÇİNDEKİLER

1. ÖZET

2. GİRİŞ

3. KOD İÇERİĞİ

3.1 Nokta Sınıfı

3.2 Doğru Parçası Sınıfı

3.3 Daire Sınıfı

3.4 Üçgen Sınıfı

3.5 Const Anahtar Kelimesi

3.6 Erişim Belirteçleri

3.7 Referans

4. SONUÇ VE ÖĞRENİLEN DERSLER

5.KAYNAKÇA

6.GİTHUB BAĞLANTILARI

1.ÖZET:

C++ dili aracılığı ile yazılmış olan bu programda nokta sınıfı, doğru parçası sınıfı, daire sınıfı ve üçgen sınıfı olmak üzere 4 adet sınıf oluşturulmuştur. Bu sınıflar içinden matematiksel hesaplar yapabilen fonksiyonlar oluşturulmuştur. Sınıfların tasarımları header dosyalarda, gerçeklenmeleri cpp dosyalarında bulunmaktadır, Sınıflar veri kapsülleme yapısına uygun olarak yazılmıştır.

2.GİRİŞ

Oluşturulan Nokta sınıfı, Doğru parçası sınıfı, Daire sınıfı ve Üçgen sınıfları ve bu sınıfların üye fonksiyonları bir sonraki bölümde açıklanacaktır.

3. KOD İÇERİĞİ:

3.1 Nokta Sınıfı:

- Nokta sınıfında 5 tane yapıcı fonksiyon oluşturulur:
 - Birincisi parametre almaz. Noktayı orijine(0,0) olarak atar.
 - İkincisi bir sayı alır ve noktanın hem x hem y koordinatına bu sayıyı atar.
 - Üçüncüsü 2 tane sayı alır. Bu sayılardan ilkinin noktanın x koordinatına atar, diğerini y koordinatına atar.
 - Dördüncüsü bir nokta nesnesi alıp onun bir kopyasını oluşturur.
 - Beşincisi bir nokta nesnesi ve 2 sayı alır. Alınan noktanın x koordinatına sayılardan ilkinin ekleyip oluşturulacak noktanın x koordinatına atar. y koordinatına sayılardan ikincisini ekleyip oluşturulacak noktanın y koordinatına atar.
- x koordinatına atama yapmak için setX(), y koordinatına atama yapmak için setY(), hem x hem y koordinatına atama yapmak için set() fonksiyonları oluşturulur.
- x koordinatına ulaşmak için getX(), y koordinatına ulaşmak için getY() fonksiyonları oluşturulur.
- Noktanın (x, y) şeklinde string gösterimini oluşturmak için toString() fonksiyonu oluşturulur.
- Nokta'yı ekrana basmak için yazdir() fonksiyonu oluşturulur. Bu fonksiyonun içinde, oluşturulan toString() fonksiyonu kullanılır.
- Tüm fonksiyonlar public, x ve y koordinatları private'tır.

3.2 Doğru Parçası Sınıfı:

- DogruParçası sınıfında 3 tane yapıcı fonksiyon oluşturulur:
 - Birincisi 2 tane nokta nesnesi alır. Bu noktalardan ilkinin doğru parçasının başlangıç noktasına atar. İkincisini bitiş noktasına atar.
 - İkincisi bir doğru parçası nesnesi alıp onun kopyasını oluşturur.
 - Üçüncüsü doğru parçasının orta noktasını(bir nokta nesnesi olarak), doğrunun uzunluğunu ve eğimini alır. Şu formülleri kullanarak başlangıç ve bitiş noktalarının koordinatlarını ayarlar:

$$P1(x1,y1) = (x0 - (L / (2 \cdot \sqrt{1 + m^2}), y0 - (m \cdot L / (2 \cdot \sqrt{1 + m^2})))$$

$$P2(x2,y2) = (x0 + (L / (2 \cdot \sqrt{1 + m^2}), y0 + (m \cdot L / (2 \cdot \sqrt{1 + m^2})))$$

- Başlangıç noktasına atama yapmak için setBaslangicNoktasi(), bitiş noktasına atama yapmak için setBitisNoktasi() fonksiyonları oluşturulur.
- Başlangıç noktasına ulaşmak için getBaslangicNoktasi(), bitiş noktasına ulaşmak için getBitisNoktasi() fonksiyonları oluşturulur.
- Doğru parçasının uzunluğunu bulmak için uzunluk() fonksiyonu oluşturulur. Şu formül kullanılır:

$$|AB| = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

- Bir kesişim() fonksiyonu oluşturulur. Bir nokta nesnesi alır, bu noktadan doğru parçasına dik olarak çizilecek doğru parçasının kesişme noktasını hesaplar.

- Doğru parçasının eğimi bulunur. Şu formül kullanılır:

$$egim = (y2 - y1) / (x2 - x1)$$

- Dik doğrunun eğimi bulunur. Şu formül kullanılır:

$$dikEgim = -1.0 / egim$$

- Doğru parçasının denklemi bulunur. Şu formül kullanılır:

$$y - y1 = egim * (x - x1)$$

- Dik doğrunun denklemi bulunur. Şu formül kullanılır:

$$y - y3 = dikEgim * (x - x3)$$

- Bu iki denklem ortak olarak çözülür ve kesişim noktası bulunur.

- Doğru parçasının orta noktasını bulmak için ortaNokta() fonksiyonu oluşturulur. Şu formüller kullanılır:

$$x0 = (x1 + x2) / 2$$

$$y0 = (y1 + y2) / 2$$

- Doğru parçasının başlangıç ve bitiş noktalarının string gösterimi için toString() fonksiyonu oluşturulur. Bu fonksiyonun içinde nokta sınıfının toString() fonksiyonu kullanılır.

- Başlangıç ve bitiş noktalarını ekrana basmak için yazdir() fonksiyonu oluşturulur. Bu fonksiyonda doğru parçası sınıfının toString() fonksiyonu kullanılır.
- Tüm fonksiyonlar public, başlangıç ve bitiş noktaları private'tır.

3.3 Daire Sınıfı:

- Daire sınıfında 3 tane yapıcı fonksiyon oluşturulur:
 - Birincisi merkez noktayı(nokta nesnesi olarak) ve yarıçapı alıp oluşturulacak dairenin merkez ve yarıçapına atar.
 - İkincisi bir daire nesnesi alıp onun kopyasını oluşturur.
 - Üçüncüsü bir daire nesnesi ve bir sayı alır. Merkez noktaya aldığı dairenin merkezini atar. Yarıçapa aldığı dairenin yarıçapının aldığı sayı ile çarpılmış halini atar.
- Yarıçapa atama yapmak için setYaricap(), merkez noktaya atama yapmak için setMerkez() fonksiyonları oluşturulur.
- Yarıçapa ulaşmak için getYaricap(), merkez noktaya ulaşmak için getMerkez() fonksiyonları oluşturulur.
- Dairenin alanını hesaplamak için alan() fonksiyonu oluşturulur. Şu formül kullanılır:

$$\text{Alan} = \pi \cdot r^2$$

- Dairenin çevresini hesaplamak için cevre() fonksiyonu oluşturulur. Şu formül kullanılır:

$$\text{Cevre} = 2 \cdot \pi \cdot r$$

- İki dairenin birbirine göre durumlarını belirlemek için kesisim() fonksiyonu oluşturulur. Parametre olarak daire nesnesi alır.
- Parametre olarak gelen daire metodu çağıran dairenin içinde ise 0, daireler birebir örtüşüyorsa 1, hiç kesişim yoksa 2 döndürür.
- İki dairenin merkezleri arasındaki uzaklık ve yarıçapları toplamı hesaplanır. Eğer merkezleri arasındaki uzaklık yarıçapları toplamından küçükse ve merkez noktaları aynıysa 1 döndürür. Aynı değilse 0 döndürür.
- Merkezleri arasındaki uzaklık yarıçapları toplamından küçük değil ise 2 döndürür.
- Dairenin merkez noktası ve yarıçapının string gösterimi için toString() fonksiyonu oluşturulur.
- Merkez noktayı ve yarıçapı ekrana basmak için yazdir() fonksiyonu oluşturulur.
- Tüm fonksiyonlar public, merkez nokta ve yarıçap private'tır.

3.4 Üçgen Sınıfı:

- Üçgen sınıfında 1 tane yapıcı fonksiyon oluşturulur. Bu fonksiyon 3 tane nokta nesnesi alır ve bunları köşe noktası olarak atar.
- Köşe noktalarına atama yapmak için setNokta1(), setNokta2(), setNokta() fonksiyonları oluşturulur.

- Köşe noktalarına ulaşmak için getNokta1(), getNokta2(), getNokta3() fonksiyonları oluşturulur.
- Köşe noktalarının string gösterimi için toString() fonksiyonu oluşturulur. Bu fonksiyonda nokta sınıfının toString() fonksiyonu kullanılır.
- Üçgenin alanını hesaplamak için alan() fonksiyonu oluşturulur. Şu formül kullanılır:

$$\text{Alan} = 1/2 \times |(x_1y_2 + x_2y_3 + x_3y_1) - (x_2y_1 + x_3y_2 + x_1y_3)|$$
- Üçgenin çevresini hesaplamak için çevre() fonksiyonu oluşturulur. Her bir kenarın uzunluğu doğru parçası sınıfının uzunluk() fonksiyonu ile bulunur. Bunların toplamı çevreyi verir.
- Üçgenin açılarını bulmak için acilar() fonksiyonu oluşturulur. Her bir kenarın uzunluğu doğru parçası sınıfının uzunluk() fonksiyonu kullanılarak hesaplanır. Her bir açı şu formüllerle hesaplanır:

$$\alpha = \arccos((b^2 + c^2 - a^2) / (2 \cdot b \cdot c))$$

$$\beta = \arccos((a^2 + c^2 - b^2) / (2 \cdot a \cdot c))$$

$$\gamma = \arccos((a^2 + b^2 - c^2) / (2 \cdot a \cdot b))$$

Burada a, b ve c kenar uzunluklarını ifade eder. Bu açılar 3 ögeli bir dizide tutulur.

3.5 Const Anahtar Kelimesi:

- Bir değerin sabit olduğunu ve değiştirilemeyeceğini belirtmek için kullanılır. Hem değişkenlerin hem de fonksiyonların tanımlanmasında kullanılabilir.

3.6 Erişim Belirteçleri:

- "public" ve "private" belirteçler, sınıfın üyelerine dışarıdan nasıl erişilebileceğini kontrol etmek amacıyla kullanılır.
- "public" belirteciyle işaretlenen üyeler, sınıfın dışından erişilebilir. Bu, sınıfın üyelerine doğrudan erişim izni verir.
- "private" belirteciyle işaretlenen üyeler, sadece sınıfın içinden erişilebilir. Sınıfın dışından doğrudan erişilmesini engeller.

3.7 Referans:

Bellek adresidir. Fonksiyonlara parametre olarak verilen referans, fonksiyonun orijinal değişkeni değiştirebilmesini sağlar.

4.SONUÇ VE ÖĞRENİLEN DERSLER:

C++ dili ile oluşturduğumuz bu proje sayesinde sınıfın tasarım ve gerçekleştirme dosyalarının ayrılmasının sağladığı avantajları görmüş olduk. Veri kapsülleme prensiplerini pekiştirdik. Sınıflar arasındaki ilişkileri anlamak, matematiksel hesaplamaları doğru bir şekilde uygulamak adına önemli bilgiler edindik.

5.KAYNAKÇA:

- https://tr.wikipedia.org/wiki/Kosin%C3%BCs_teoremi
- <https://www.matematikrehberim.com/noktanin-analitigi.php?id=60>

6.GİTHUB BAĞLANTILARI:

Hilal Aydın:

<https://github.com/HilalAydinn>

Zehra Kandaz:

<https://github.com/ZehraKandaz>